molecules

MDPI

Article

Molecular Property Prediction of Modified Gedunin Using Machine Learning

Mohammed Aly [1,*] and Abdullah Shawan Alotaibi [2]

[1] Department of Artificial Intelligence, Faculty of Artificial Intelligence, Egyptian Russian University, Badr City 11829, Egypt
[2] Computer Science Department, Shaqra University, Shaqra City 11961, Saudi Arabia
* Correspondence: mohammed-alysalem@eru.edu.eg

Abstract: Images of molecules are often utilized in education and synthetic exploration to predict molecular characteristics. Deep learning (DL) has also had an influence on drug research, such as the interpretation of cellular images as well as the development of innovative methods for the synthesis of organic molecules. Although research in these areas has been significant, a comprehensive review of DL applications in drug development would be beyond the scope of a single Account. In this study, we will concentrate on a single major area where DL has influenced molecular design: the prediction of molecular properties of modified gedunin using machine learning (ML). AI and ML technologies are critical in drug research and development. In these other words, deep learning (DL) algorithms and artificial neural networks (ANN) have changed the field. In short, advances in AI and ML present a good potential for rational drug design and exploration, which will ultimately benefit humanity. In this paper, long short-term memory (LSTM) was used to convert a modified gedunin SMILE into a molecular image. The 2D molecular representations and their immediately visible highlights should then provide adequate data to predict the subordinate characteristics of atom design. We aim to find the properties of modified gedunin using K-means clustering; RNN-like ML tools. To support this postulation, neural network (NN) clustering based on the AI picture is used and evaluated in this study. The novel chemical developed via profound learning has long been predicted on characteristics. As a result, LSTM with RNNs allow us to predict the properties of molecules of modified gedunin. The total accuracy of the suggested model is 98.68%. The accuracy of the molecular property prediction of modified gedunin research is promising enough to evaluate extrapolation and generalization. The model suggested in this research requires just seconds or minutes to calculate, making it faster as well as more effective than existing techniques. In short, ML can be a useful tool for predicting the properties of modified gedunin molecules.

Keywords: CNN; LSTM; RNN

check for updates

Citation: Aly, M.; Alotaibi, A.S. Molecular Property Prediction of Modified Gedunin Using Machine Learning. *Molecules* **2023**, *28*, 1125. https://doi.org/10.3390/molecules28031125

Academic Editor: Erich A. Müller

Received: 8 December 2022
Revised: 16 January 2023
Accepted: 18 January 2023
Published: 23 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The prediction of molecular characteristics in the realm of drug development is an important effort. Computer approaches, due to their accurate forecasting, can speed up the whole interaction between finding better applicants in a faster, cheaper manner. This is particularly convincing when you consider that the usual cost of improving a drug is now estimated to be approximately USD2.8 billion [1]. The usual technique of using silicon for predicting molecular characteristics depends mostly on the removal of fingerprints or hand-made highlights, which are subsequently used in combination with AI computation. In conclusion, this kind of atomic portrayal is unilaterally presented by space professionals, to gather the highlights essential for the task to be carried out [2]. To advance beyond that type of tendency to a broader approach, several forms of IAs were incorporated into the field of prediction of molecular properties. The calculation of deep learning has resurged in particular, due not only to the acceleration of computer

power and the increase in accessibility of large information indices but also to the vast scale of its performance, for instance, in the field of characteristic language use [3] and example recognition [4]. These networks are mechanized to study representations for a certain task and can thus eliminate the confused design interaction of the components [5]. A suitable portrait for molecules should be created to use deep learning calculations and cover the explicit component design region. As molecules may be seen as diagrams, one technique just involves using an atomic graph portrait—resulting in the improvement of neural graphic networks (GNNs) which have become increasingly commonplace in recent years [6–8]. The achievements in the development of traditional AI techniques, in particular, with expectations of quantum mechanical properties [5,9–12], physicochemical properties, as well as hydrophobicity, [13–15] and predictions of toxicity, appear to give perhaps the most encouraging profound learning strategies in explicit diagram tasks. Due to the rapid speed growth of the distributions recognized with GNN expectations for molecular properties, it may sometimes be difficult to know the current situation in this sector.

A deep study into synthetic scanning becomes more important at a speedy rate and when the prediction of particle characteristics, such as free solvent energy [16] or ionizing energy [17], contends with or even outstrips existing abdominal muscle calculations. The range of potential AI (ML) calculation errant appears endless and can range from a grouping of bioactive mixtures into dynamic and non-dynamic HIV replication classifications [18] to retrograde assignments in which a particular value, such as lipophilicity, is predicted, depending on the information [19]. However, ML computations are not limited to class and quality forecasts. They can generate new compound buildings depending on input structures or desired communication goals. They can thus maintain the plan of new molecules in the fields of materials and restorative chemistry [20–22]. Clustering further evaluates the partitioning of huge nuclear datasets into the most highly harmonic collections and can be utilized from therapeutic research to identify innovative lead structures to speeding up the selection of your PC [23]. Clustering of K-Means is an unmonitored learning-technique grouping the unmarked data into several clusters. Here, K determines the number of predefined clusters that have to be generated, if K = 2, two clusters would be created, and for K = 3, three clusters would be created, and so on.

The usage of strings is a simple way of addressing a subatomic design. A string is a number and letter progression that is used routinely to enhance the determination of subatomic information line transit (SMILES). In these strings, the SMILES string 'CCCO' would be used to depict each particle with its edge iotas and neighboring bonds, and an atom such as n Propanol [24]. As the chirality was not properly represented, the IUPAC offered an alternative string depiction called the International Chemical Identifier (InChI) which could not be easily interpreted by a chemist, apart from the focused chiral and the tautomerism [25].

Sub-atomic descriptors are another method to address a sub-atomic design. These descriptors may be direct, directly available from the construction, with highlights such as the number of iotas, the number of heteroatoms, or the subatomic load to establish or reconstruct real characteristics such as the second dipole and the incomplete loading dramatically [26]. These characteristics are then combined into a fixed-length vector in a described request. Additional features such as subatomic fingerprints are available and are typically used. An atomic fingerprint is a longitudinal vector that deals with each component.

In the study of Tang et al. [27], the suggested model was used to predict AASC compressive strength. The ANN model took into account the complicated and nonlinear relationship between compressive strength and concrete components, such as raw materials used in the preparation of the concrete specimen. The relation between both variables and the AASC compressive strength was inverse, with strength increasing but decreasing with a rise in SS. The provided model was further justified as exhibiting high efficiency in predicting strength properties by comparing it to current models.

In the study of Yakub et al. [28], the suggested model was used to discuss the impacts of crystallite size on the performance of copper and iron oxides in reducing NOx, as well as the construction of a prediction model that links crystallite size with H2-SCR efficiency. Monometallic as well as bimetallic catalysts doped over palm kernel shell-activated carbon were explored. Aside from chemical analysis, two predictive equations for NOx conversion and N2 selectivity were established using the ANN approach. The presented model was most relevant for working temperatures ranging from 250 to 300 °C, whereas lower temperature ranges would need additional data.

In the study of Abhyankar et al. [29], the suggested model employed Envisat ASAR VV polarized data and an ANN to detect flooded regions caused by a cyclonic storm. This study solely covered nine mandals in the Guntur district of Andhra Pradesh to identify the flooded regions. The outcomes revealed a high degree of classification accuracy and implied that this might be a quick tool for damage estimates and post-disaster relief and recovery activities.

In the study of Etu et al. [30], the accuracy of the Radial Basis Function Neural Network (RBFNN) as well as the Multiple Linear Regression model (MLR) in forecasting home-based visits was evaluated. The datasets for the study were obtained from a household travel survey in Akure, Nigeria, which used the SPSS 22.0 statistical tool. This study proved the better accuracy of the RBFNN in creating trip-generation forecasts in the study region, and as a result, it was suggested for scientists in carrying out such forecasts.

2. Results

LSTM was used to convert the modified gedunin SMILE into an image, which was further classified in cluster 2. In this paper, Curve Elbow was applied to determine the number of clusters (Figure 1). Similar property compounds in cluster 2 included chembl id, molecular weight, molecular formula, and SMILE (Figure 2). This classification is executed by K-means clustering. Based on a silhouette score, four clusters were determined to split molecules into clusters based on common functions or properties using features of a large dataset (Figure 3) as a training and validation set. Modified gedunin shares properties with 933 compounds as shown in (Figure 2). Properties are interleukin-23 receptor inhibitor, adenosine kinase and mitotic spindle assembly protein MAD2B inhibitor, antiproliferative activity, cytotoxicity against human HCC366 cells, anticancer activity against human T47D cells, and growth inhibition of human MCF7 cells, antiprogestational activity, anti-inflammatory activity and anti-tumor activity.

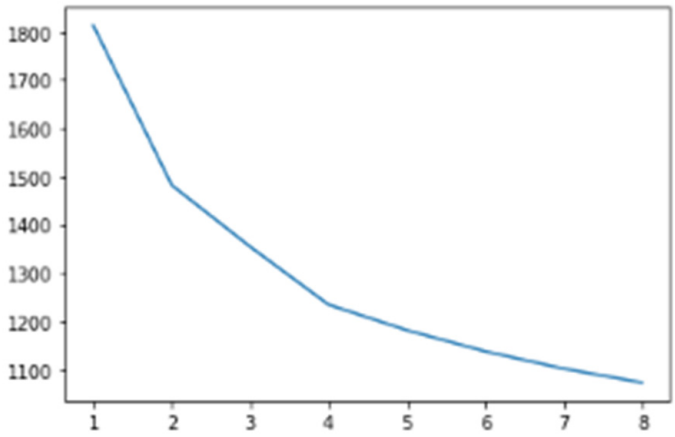


Figure 1. Curve Elbow to determine number of clusters.

A	B	C	T	U	V
	ChEMBL ID	Molecular Weight	Molecular Formula	Smiles	Catego
4	CHEMBL4459526	324.79	C13H13ClN4O2S	O=C(Cl)CCCCC(=	2
8	CHEMBL4448005	344.84	C19H21ClN2O2	O=C(NC1CCCNC1	2
19	CHEMBL169111	330.8	C16H11ClN2O2S	[O-][S+]1Cc2c(nr	2
25	CHEMBL4446013	335.36	C19H17N3O3	CC(=O)Nc1cccc(-	2
27	CHEMBL225696	328.42	C21H20N4	Cc1ccc(-c2c(N)nc	2
42	CHEMBL4448006	336.34	C20H16O5	CC1=C(C)C2=CC(	2
43	CHEMBL4468545	333.37	C16H15NO5S	COc1cc2sc(C(=O)	2
49	CHEMBL132713	300.36	C20H16N2O	OC1(c2cccc3cccc	2
51	CHEMBL1215059	325.8	C14H20ClN5O2	CC(C)CN(NC(=O)	2
55	CHEMBL3929427	308.34	C18H16N2O3	CCn1c(C#N)ccc1-	2
59	CHEMBL4520271	342.26	C13H10F4N6O	Fc1ccccc1Nc1nc2	2
64	CHEMBL4461210	326.77	C12H11ClN4O3S	O=C(CCCCl)Nc1n	2
71	CHEMBL1684100	334.39	C21H19FN2O	O=C1c2ccc(C#Cc3	2
75	CHEMBL132976	346.44	C15H14N4O2S2	CCOC(=O)c1cnc2	2
80	CHEMBL3230400	314.86	C19H23ClN2	Cc1ccc2c(c1)[C@	2
88	CHEMBL421491	304.43	C19H28O3	CC12CCC3C(C(=C	2
92	CHEMBL1254929	313.4	C19H23NO3	COC(=O)[C@H]((	2
99	CHEMBL447280	308.34	C11H8N4O3S2	COc1ccc2[nH]c(S	2
103	CHEMBL209098	302.46	C19H30N2O	CCN1CCc2cccc(O	2
107	CHEMBL3427326	323.44	C20H25N3O	CCCOc1cccc(/C=N	2
120	CHEMBL1091507	349.48	C17H19NO3S2	Cc1ccc(C(C)/C=C	2
124	CHEMBL1094204	310.39	C17H26O5	O=C(O)C[C@H]1	2
126	CHEMBL2178003	315.32	C13H12F3N3OS	Cn1nc(C(F)(F)F)c	2
128	CHEMBL4449699	347.42	C20H21N5O	Cc1cc(-c2cccc(NC	2

final_frame_Tues - Copy

933 of 5017 records found

Figure 2. Similar property compounds in cluster 2.

```
df_1 = pd.read_csv('C:\\Users\\PRIYA DAGAR\\Desktop\\all folder\\ml project\\data\\Research Files\\compound_1.csv')
```

```
df_1.head()
```

CX .ogP	CX LogD	Aromatic Rings	Heavy Atoms	HBA Lipinski	HBD Lipinski	#RO5 Violations (Lipinski)	Molecular Weight (Monoisotopic)	Molecular Species	Molecular Formula	Smiles
3.26	0.14	0.0	23.0	4.0	1.0	0.0	320.1988	ACID	C19H28O4	O=C(O)C[C@H]1CC[C@]2(CC1)OC[C@]1(O2)C2CC3CC(C2...
1.58	0.11	1.0	24.0	7.0	1.0	0.0	331.1532	ACID	C17H21N3O4	CC1=NC(C)=C([N+](=O)[O-])C(c2cccc(C)n2)C1=C(/...
1.15	-0.34	1.0	23.0	7.0	1.0	0.0	317.1376	ACID	C16H19N3O4	CCOC(C(O)=C1/C(C)=NC(C)=C([N+](=O)[O-])C1c1cccc...
3.82	3.82	3.0	25.0	4.0	0.0	0.0	339.1271	NEUTRAL	C20H18FNO3	C[C@H](COc1ccccc1)Oc1ccn(-c2ccc(F)cc2)c(=O)c1
1.72	1.21	2.0	21.0	6.0	1.0	0.0	324.0448	NEUTRAL	C13H13ClN4O2S	O=C(Cl)CCCCC(=O)Nc1nnc(-c2ccncc2)s1

```
df_2 = pd.read_csv('C:\\Users\\PRIYA DAGAR\\Desktop\\all folder\\ml project\\data\\Research Files\\protein_target.csv')
```

```
df_2.head()
```

	ChEMBL ID	Molecular Weight	AlogP	PSA	HBA	HBD	#Rotatable Bonds	CX ApKa	CX BpKa	CX LogP	CX LogD	Aromatic Rings	Inorganic Flag	Heavy Atoms	HBA Lipinski	HBD Lipinski	Molecul Weigl (Monoisotop
0	CHEMBL2372345	343.38	-0.71	147.82	6	4	7	3.67	8.23	-2.95	-3.01	0	-1	24	9	5	343.174
1	CHEMBL187443	337.38	-0.40	141.75	5	5	8	3.98	7.73	-2.25	-2.39	1	-1	24	8	6	337.160
2	CHEMBL370056	436.51	-0.60	137.21	6	3	7	10.73	0.00	-0.70	-0.70	0	-1	31	10	3	436.232
3	CHEMBL2371249	321.38	-0.10	121.52	4	4	8	4.07	7.84	-1.94	-2.06	1	-1	23	7	5	321.166
4	CHEMBL1221522	369.44	-0.30	141.75	6	5	10	3.93	7.73	-2.48	-2.63	1	-1	25	8	6	369.135

Figure 3. Dataset for training and validation.

We used principal component analysis (PCA) to illustrate the ten properties (interleukin-23 receptor inhibitor, adenosine kinase, and mitotic spindle assembly protein MAD2B inhibitor, antiproliferative activity, cytotoxicity against human HCC366 cells, anticancer activity against human T47D cells, growth inhibition of human MCF7 cells, antiprogestational activity, anti-inflammatory activity, and anti-tumor activity). As shown in Figure 4, for every one of the molecules created by the LSTM as well as randomized molecules from the training set, four molecular descriptors have been examined. To display the data, PCA was performed for dimension reduction, with produced molecules in red as well as training data molecules in blue. The distributions are nearly identical.

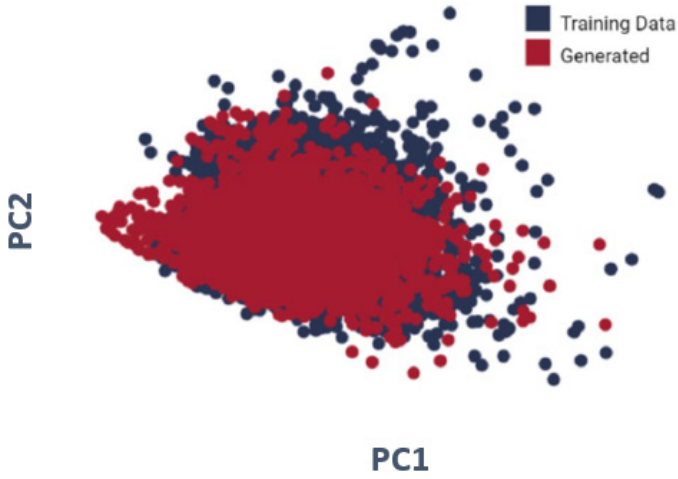


Figure 4. The PCA projection of molecular descriptors from training data as well as molecules created by the LSTM.

The LSTM was utilized to produce the SMILE of modified gedunin. Furthermore, the LSTM focuses on the most optimum region of the chemical space. We also separately displayed the distributions of the ten attributes. Molecules formed in the last transfer-learning iteration have minimized the objective values to levels significantly lower than molecules generated before transfer learning; hence, the model not only concentrates on, but also identifies new, optimum parts of the chemical space.

Several ML models that use various algorithms detect images with high accuracy but become ineffective when processing large quantities of data. To classify the target cell, initially train the ML model so that it can recognize the cell as well as its features, which is achieved by contrasting the image of the targeted cells, which isolates it from the background. Having sufficient data is critical to the accuracy of prediction models. Given a variety of parameters, neural models need a large quantity of training data in order to acquire an appropriate chemical representation and, eventually, the prediction objective itself. Accuracy is used in classification problems to assess the percentage of correctly predictions made by a model. In ML, the accuracy score is an assessment metric that relates the number of correct predictions by a model to the overall number of predictions made. Equation (1) is used to measure the accuracy by dividing the number of correct predictions by the total number of predictions [31,32]:

$$\text{Accuracy} = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad (1)$$

The total accuracy of the suggested model was 98.68%. The accuracy of the molecular property prediction of modified gedunin research is promising enough to evaluate extrapolation and generalization.

3. Materials and Methods

3.1. Data Collection and Preprocessing

With the CHEMBL and Drug Bank datasets, all relapse businesses were conducted and the Python code under the Rkdit environment was used to transform the SMILES texts to graphic pictures as part of the data pre-processing. ChEMBL is an open data resource that contains information on binding, functional, and ADMET properties for a large number of drug-like bioactive chemicals. On a routine basis, these data are manually extracted from the main published literature, then selected and standardized to enhance their quality and usability across a wide range of chemical biology as well as drug-discovery research concerns. The CHEMBL database now includes 5.4 million bio-activity measurements for over 1 million chemicals as well as 5200 protein targets. The web-based interface, data downloads, and online services are accessible at: https://www.ebi.ac.uk/chembldb (accessed 20 March 2022). The SMILES string format was used to identify molecules for simple interpretation by the RNN model we use. SMILES was built with grammatical consistency as well as machine friendliness in mind, with characters representing atoms, bonds, and chemical structures (Figure 5) [33]. These molecules ranged in length from 35 to 75 characters. These SMILES molecules were encoded in a single pass so that each SMILES character has been defined by a 53-dimensional vector of zeros with a one in the relevant index of the character. These data were then applied to train an RNN, which was then applied to construct acceptable molecules.

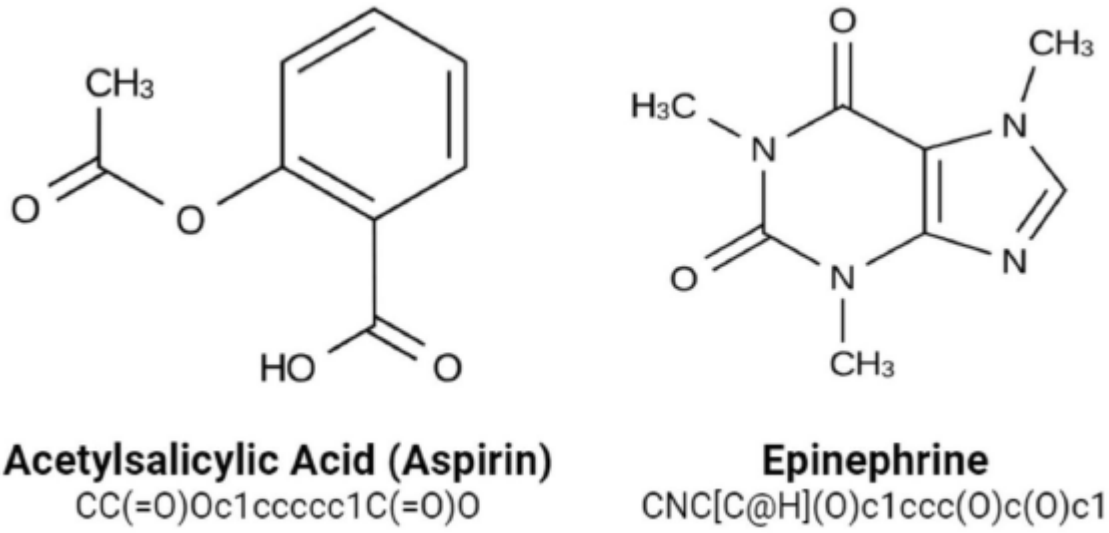


Figure 5. SMILES notations for numerous molecules.

3.2. Feature Extraction and Clustering

VGGNet is a revolutionary neural network design, that was exhausting features and was proposed by Andrew Zisserman and Karen Simonyan at Oxford University in 2014. In order to cluster data in needed groupings based on a similarity index, the K-mean clustering method was employed. Using the Elbow technique, the cluster number was determined.

We only used this model as a feature extractor, which means we removed the last (prediction) layer to generate a feature vector. We specified the target size to (53, 53) when loading the molecular images since the VGG model requires the molecular images it receives to be 53 × 53 NumPy arrays. We were then able to manually load the VGG model as well as delete the output layer. This indicated that the new final layer was completely connected. Now that the last layer was removed, we used the prediction technique to obtain our feature vector. We utilized this feature_extraction function to extract features from all of the molecular images and saved them in a dictionary. Then, we reduced the dimensions of the feature vector to a much smaller number. We clustered our molecular images next because we had a reduced feature set. The K-mean clustering approach would

then enable us to classify our feature vectors into k clusters (k = 4). Therefore, every cluster must contain related molecular images. This allowed us to classify the molecular images into clusters. We were able to see that our model did a good job of clustering the molecular images.

3.3. Molecular Property Prediction of Modified Gedunin

The properties of modified gedunin were anticipated based on the similarity index by predicting the cluster it fits. The idea of virtual screening has fueled decades of study into ML approaches for predicting molecular characteristics varying from bioactivity to bio-distribution and physical features. If effective, these approaches have the potential to change the discovery process by decreasing the time and cost involved with experimental screening while expanding the chemical space that may be investigated. Although traditional testing screening methods may potentially examine millions of molecules, virtual screening may evaluate billions in a short amount of time. Comparable to how medicinal chemists evaluate molecules, the techniques employed in virtual screening try to understand the link between distinct chemical substructures and the target features. The idea is that when given huge quantities of training data in the form of molecules with detected properties, these techniques will be able to systematically catch complex features that people may miss. Figure 6 illustrates an outline of machine learning techniques for predicting molecular properties. A training dataset of molecules and accompanying data is applied to design a model that predicts a collection of test observables.

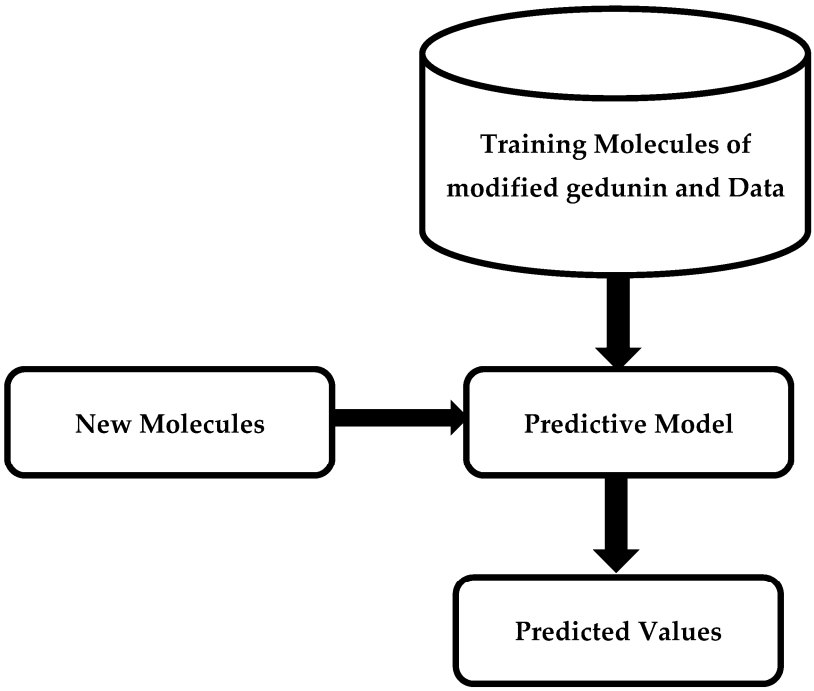


Figure 6. A diagram illustrating the predictive modelling method.

3.4. Neural Networks for Property Prediction

Advances in neural models have created the opportunity of changing this current state of the art. One of the most significant differences between these models is how molecules are handled within the ML technique. The accuracy of the extracted features, which might not be ideal for a specific predictive modeling purpose, is critical to the performance of these models. Neural models, on the other hand, learn their expert feature representations directly from data, converting a molecular graph into a dense continuous vector. Graph convolution, a method borrowed from earlier research in image processing, is indeed a frequent strategy for such translation. While this vector's coordinates do not often capture

human attributes, they have already been proven to be quite adaptable and capable of capturing complicated relations given enough data. The Tox21 Challenge, in which groups employed a range of ML algorithms to predict the outputs of experimental toxicity assays, was an early demonstration of DL effectiveness in drug development.

3.5. Data Requirements for Accurate Property Prediction

Access to sufficient data is critical to the accuracy of prediction models. Assuming a wide range of parameters, neural models need a large quantity of training data in order to acquire an appropriate chemical description and, eventually, the prediction problem itself. Unfortunately, obtaining a high sample size may not be possible in many drug-development settings. A regular pharma-lead optimization attempt, which produces only a few thousand molecules does not always create enough data to train a neural model. Furthermore, even a huge training set does not really ensure that the model will effectively adapt to a new drug space. The greater the difference from the initial training set, the more difficult it is for the model to detect the goal attribute properly. The available approaches for estimating confidence cannot accurately estimate the possibility of the model prediction being right.

3.6. Uncertainty Quantitation

When an ML model is applied to identify a molecule's biological activity or physical attributes, it is essential to recognize the prediction's uncertainty. Although it is reasonable to assume that predictions on molecules identical to those in the training set might be more definite, the field is still unable to agree on effective techniques for quantifying uncertainty. Even as the field progresses toward more complicated learned representations, such as those outlined in Section 3.4., quantifying molecular similarities as well as assessing uncertainty in ML models becomes more complex. Hirschfeld et al. [34] evaluated multiple strategies for uncertainty quantification in NN models using 5 benchmark datasets. The ability of numerous established methodologies to quantify the inaccuracy in model predictions was investigated by these authors. The authors reported that none of the strategies studied could estimate uncertainty consistently across datasets.

3.7. Recurring Neural Network (RNN)

RNNs have demonstrated success in modeling sequential data, which is typically encountered in natural language processing (NLP). RNNs may recognize the meaning of data in addition to capturing the data's grammatical structure. RNNs can always be thought of as several clones of the exact same NN, each one sending input to its successor via a hidden state. By factoring in this hidden state, each NN provides probabilities to the next item in the sequence given all those that came before it. Assuming network parameters ρ, the probability that the whole sequence $S = s_1\ s_2\ s_3 \ldots\ s_N$ of size N time steps is as follows:

$$P_\rho(S) = P_\rho(s_1)P_\rho(s_2|s_1)P_\rho\ (s_3|s_1s_2) \ldots P_\rho\ (s_N\ |s_1 \ldots s_{N-1})$$

Recent neural networks use training data to learn, such as feedforward and convolutionary neural networks (CNNs). They are characterized by their "memory" since they use previous inputs for information to impact the present input and output whereas the artificial recurrent neural network (RNN) architecture utilized in deeper learning is the long short memory (LSTM). In contrast to the typical neural networks for feedforward, LSTM contains feedback connections. Not only single data points (such as pictures) but also whole data sequences may be processed (such as speech or video). LSTM is appropriate for applications such as unsegmented, connected handwriting, voice recognition, and network traffic anomaly or IDS detection (intrusion detection systems), and the visual geometry group VGGNet architecture is ILSVR2014's first classifying architecture whereas GoogLeNet is the winner. This explains why VGGNet is built up on the top of this architecture by many contemporary model classifications.

3.8. Long Short-Term Memory (LSTM)

We employed an RNN with LSTM. The LSTM network is a kind of RNN designed to simulate long-term relationships appropriately. LSTMs are made up of cells, each of which has three neural network layers known as gates (forget, update, and output gates). These gates decide which data to keep in an extra cell-state. The cell state traverses the whole network; hence, the hidden state of an LSTM serves as short-term memory, although the cell state serves as long-term memory. LSTM was used to generate SMILE of modified gedunin (Figure 7). As shown in (Figure 8) to begin, the character "G" was entered, which started up the hidden and cell states. The network started sampling symbols one by one until the last character, "\n," was created. To generate the output logits, a dense layer was added after the LSTM cells. After the LSTM cells, a dense layer was utilized to generate the output logits, after which they were transformed to probabilities using a Softmax pro 7.1 layer throughout sampling. The model was built with Pytorch, a major Python ML package. During training, molecules were sampled from the model to assess progress (Figure 8); the model learned fast to create acceptable molecules. The amount of data utilized to train neural models determined their performance. It is feasible to augment data collection in some ML fields by altering training samples.

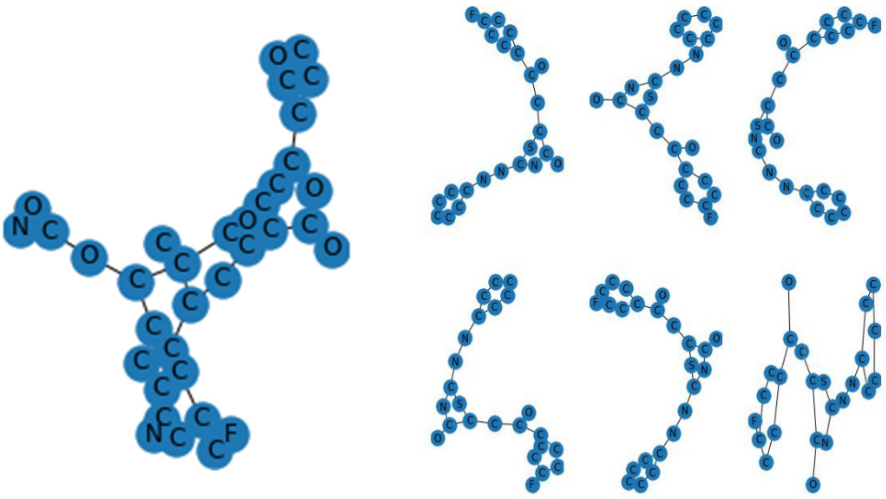


Figure 7. Molecular image of newly generated SMILES using LSTM.

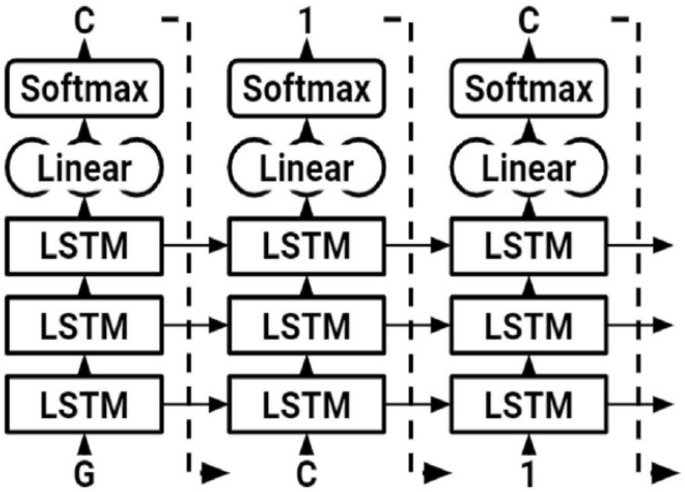


Figure 8. LSTM used to generate SMILE of modified gedunin.

To validate that the model was running in the exact chemical space as the initial training data, we examined the molecules created by the LSTM with those found in the initial training data. The features of the molecules in the given dataset and those created by the LSTM overlapped considerably, demonstrating that the model could correctly replicate but not exactly copy the training data. To illustrate the ten properties (interleukin-23 receptor inhibitor, adenosine kinase, and mitotic spindle assembly protein MAD2B inhibitor, antiproliferative activity, cytotoxicity against human HCC366 cells, anticancer activity against human T47D cells, growth inhibition of human MCF7 cells, antiprogestational activity, anti-inflammatory activity, and anti-tumor activity), we used principal component analysis (PCA).

LSTM does not have critical connections such as conventional feed for neural organizations. It can not only cycle single-focused information (such as photos), but also entire information groups (such as discourse or video). For example, LSTM is suitable for messages such as unsegmented penalty recognitions, speech identification, and detection of an anomaly in network traffic or IDSs (interruption location systems).

3.9. Convolutional Neural Network (CNN)

CNNs are fundamental neural organizations, which are extremely layered, and the large majority of them feature comparable essential capacities, including convolution layers, pooling, and grouping layers (Figure 9). CNNs basically contrast each other by introducing and grouping these key layers and also by developing an organizing plan. In the first place, the information images are specifically pre-processed as standard. The input stream is then transferred into many convolutionary layers with pooling layers in which extraction and recurrence are highlighted. The basic features build continuously in a constructive way. Each highlight is afterward partially consolidated and the following highlights each signify a design aspect of the class called long term; these highlights are transferred to the totally related layer and this layer gives a measurement of the order.

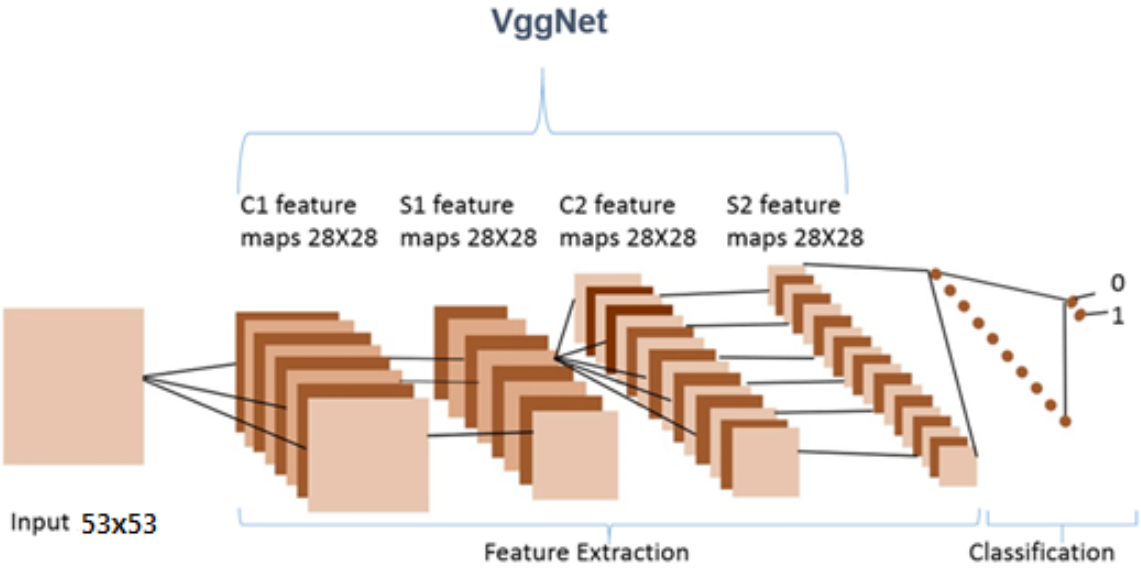


Figure 9. The architecture of CNN layers for feature extraction and classification [35]. The classifier is typically made up of fully connected layers.

3.9.1. Pre-Processing Layer

Several information images must be loaded from the information layer. A few pre-processing activities, including measurement and standardization, are necessary before this. However, CNNs ask for far fewer pre-processing tasks than other neural networks. Another important factor in removing uninteresting contrasts is the basic pre-processing layer.

3.9.2. Convolutional Layer

By eliminating interesting highlights from it, a convolutionary layer consolidates the contribution, including maps due to different component locators, as illustrated before. In the first convolutionary layer, the neurons scrub straight as edges. In the accompanying convolutionary layers, neurons can collect data to obtain a larger image of the image, therefore making a high-demand distinction between them [36,37]. Every turning bite is equipped for highlighting the removal across the information plane, although neurons are allocated to diverse parts of the information photographic plane to provide the part maps in the same size of the responsive field.

3.9.3. Convolution

Each convolutionary layer has several limits such as the information size, bit size, guide stack depth, zero coiling, and step.

3.9.4. Activation

In all cases, an actuation should occur after the weighted whole and a propensity. Apart from pure perceptrons, the simple direct mix of information is predicted to break apart and make it viable for a neuronal organization to become the general approximation of non-stop capabilities in a Euclidean environment. However, Jarrett et al. eventually brought the Corrected Linear Units (ReLUs) to CNNs to enhance the display. For some time now, Xavier Glorot et al. have pointed out that the very strong non-linearity, differentiability, and insufficient component of ReLU must be acknowledged. Ultimately, ReLUs are generally recognized for initiating convolutional yields.

3.9.5. Pooling Layer

Pooling or sub-sampling includes several kinds of activities such as wide pooling, pooling, and so on; pooling usually interferes with a few convolutionary levels [38]. The most-often used pooling techniques in CNNs are max-pooling and normal pooling. Pooling offers CNNs several benefits. In particular, grouping goals to prevent overcrowding by concentrating information nearby with a window that decreases the dimensionality of the information. Dimensional information decreases further assistance to reduce estimates. In addition, pooling achieves invariance including interpretation, pivot, and scale since a few distinctions or scaling do not distinct after appropriate pooling.

3.9.6. Classification Layer

The group layer is the top tier of the organization, where the final enmeshed element is collected and a segment vector is returned, where each column is directed to a class. Most clearly, each component of the yield vector deals with the probability assessment for each class and the number of components [39]. While convolutions and information about the crude map in the element space are linked, the arrangement layer results in an example space presentation, which offers a remarkable exhibition of order. Unique CNNs generally end up with a fully related layer. The entirely linked characterization layer is a heritage, which goes out to AI with "maintain extraction and grouping". All high requirements are joined and weighed by complete connections of neuronal contributions to achieve the spatial change.

3.10. Transfer Learning

A deep CNN model was developed using transfer learning to increase CNN's expectation accuracy (Figure 10). The chembl dataset was used for the pre-workout of the model. The model was pretrained on 5021 compounds and the particular atomic characteristics were removed. After the preparation of the scenes, a flattened layer trapped through a dense layer, a drop-out layer, and a single node dense cap were used to shorten and subtract the pre-trained models. LSTM was used to convert the modified gedunin SMILE into an image, which was further classified in cluster 2 (Figure 11)

```
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_6 (InputLayer)         [(None, 180, 180, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 180, 180, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 180, 180, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 90, 90, 64)        0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 90, 90, 128)       73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 90, 90, 128)       147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 45, 45, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 45, 45, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 45, 45, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 45, 45, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 22, 22, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 22, 22, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 22, 22, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 22, 22, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 11, 11, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 11, 11, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 11, 11, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 11, 11, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 5, 5, 512)         0
_________________________________________________________________
global_average_pooling2d_5 ( (None, 512)               0
=================================================================
Total params: 14,714688
Trainable params:14,714688
Non-trainable params: 0
```

Figure 10. Transfer Learning vgg16 was applied for molecular feature extraction.

	CHEMBLID	Category
0	CHEMBL307662	1
1	CHEMBL2063796	2
2	CHEMBL2418293	0
3	CHEMBL3278821	0

Figure 11. New molecule created with clusters assigned.

3.11. K-Means Clustering

In this paper, K-means was applied to identify molecular clusters. Let us suppose that we needed to visualize the M layers of VGG16. The output of this layer could have the shape $(A,\ B,\ C)$. As a result, we obtained $Z * C$ dimensional vectors (for one specific image), where $Z = A * B$. We can already implement K-Means to these vectors (four clusters) and color this image based on the clustering result. Because the $A * B$ feature map indicates a scaled-down version of this image, we simply colored the corresponding image region for each of these vectors.

The K-Means partition algorithm is a bunch-like method and J.B. MacQueen proposed it. This solo algorithm is typically used in the mining of information and for example recognition. This algorithm is defined as targeting group execution files, square error, and error. This method tries to find K divisions to show a certain standard to seek the optimization outcome. Right off the bat, a few spots were selected to address the central points of the underlying group (usually, the primary K pay points are selected in a case for the underlying bunch point of convergence); furthermore, the remaining example dabs were assembled in the middle of the central focus, based on the lower distance, at that point, we obtained the underlying arrangement. K-Means isolation-dependent algorithm is a kind of bunch algorithm and benefits from speed, skill, and speed. This technique, however, depends greatly on introductive specks and the difference to start selecting instances, which always yields different results. In addition, this track-dependent algorithm uses the slope method constantly to get extremely high. The search is constantly performed in the angle technique along the direction where energy is diminished, which results in the lesser point of the neighborhood when the underlying convergence point is not legitimate. A total of 4 clusters were chosen to divide molecules into clusters based on comparable functions or characteristics based on a silhouette score (Figure 1). New molecules were created in the shape of SMILES and allocated clusters in Figure 11.

4. Conclusions

In this work, we applied an LSTM with RNNs to predict the properties of molecules of modified gedunin. According to the outcomes and analysis, it is clear that DL (CNN, transfer learning) and K-means algorithm can be applied to predict properties of molecules of newly generated compounds using a SMILE as raw data. More than 5000 molecules for training were utilized and vggnet for extraction of features was employed in this paper. Based on a resemblance of characteristics based on LSTM/RNN, clusters were developed with pre-exercise data and the newly generated SMILE of modified gedunin. During training, molecules were sampled from the model to assess progress; the model learns fast to create acceptable molecules. The quantity of data utilized to train neural models determine their performance. To validate that the model was running in the exact chemical space as the initial training data, we examined the molecules created by the LSTM with those found in the initial training data. The features of the molecules in the given dataset and those created by the LSTM overlap considerably, demonstrating that the model can correctly replicate but not exactly copy the training data.

Our current study extends the use of ML to predict the molecular properties of modified gedunin. To support this postulation, neural network clustering based on the AI picture was used and evaluated in this study. The novel chemical developed via profound learning has long been predicted on characteristics. As a result, LSTM with RNNs allows us to predict the properties of molecules of modified gedunin. In addition, every iteration of transfer learning results in a PCA projection of the molecular descriptors of created molecules. To optimize the required features, the model concentrates on and begins to uncover new areas of the chemical space. The chembl dataset was used for the pre-workout of the model. The model was pre-trained on 5021 compounds and the particular atomic characteristics were removed. After the preparation of the scenes, a flattened layer trapped through a dense layer, a drop-out layer, and a single node dense cap were used to shorten and subtract the pre-trained models. The total accuracy of the suggested model is 98.68%.

It is obvious that the accuracy of the molecular property prediction of modified gedunin research is promising enough to evaluate extrapolation and generalization.

The model suggested in this research requires just seconds or minutes to calculate, making it faster as well as more effective than existing techniques. In short, ML can be a useful tool for predicting the properties of modified gedunin molecules.

The Cheminformatics app used to conduct these analyses is in ongoing development and is still in pre-release form. In the future, new models and functionality will be introduced.

Author Contributions: Conceptualization, M.A.; methodology, M.A.; formal analysis, M.A.; investigation, M.A. and A.S.A.; data curation, A.S.A. and M.A.; software, M.A.; writing—original draft preparation, M.A. and A.S.A.; writing—review and editing, M.A.; visualization, A.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable. The authors have no competing interests to declare that are relevant to the content of this article.

Data Availability Statement: The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest. The authors have no competing interests to declare that are relevant to the content of this article. The submitted work is original and has not been submitted to another journal for simultaneous consideration. The manuscript is not published elsewhere in any form or language.

Sample Availability: Samples of the compounds are not available from the authors.

References

1. Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; et al. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **2019**, *59*, 3370–3388. [CrossRef] [PubMed]
2. Merkwirth, C.; Lengauer, T. Automatic generation of complementary descriptors with molecular graph networks. *J. Chem. Inf. Model.* **2005**, *45*, 1159–1168. [CrossRef] [PubMed]
3. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing [review article]. *IEEE Comput. Mag.* **2018**, *13*, 55–75. [CrossRef]
4. Bhamare, D.; Suryawanshi, P. Review on reliable pattern recognition with machine learning techniques. *Fuzzy Inf. Eng.* **2018**, *10*, 362–377. [CrossRef]
5. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017. [CrossRef]
6. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A comprehensive survey on graph neural networks. *arXiv* **2019**, arXiv:1901.00596. [CrossRef]
7. Zhou, K.; Dong, Y.; Lee, W.S.; Hooi, B.; Xu, H.; Feng, J. Effective training strategies for deep graph neural networks. *arXiv* **2020**, arXiv:2006.07107.
8. Mayr, A.; Klambauer, G.; Unterthiner, T.; Steijaert, M.; Wegner, J.K.; Ceulemans, H.; Clevert, D.-A.; Hochreiter, S. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chem. Sci.* **2018**, *9*, 5441–5451. [CrossRef]
9. Wu, Z.; Ramsundar, B.; Feinberg, E.N.; Gomes, J.; Geniesse, C.; Pappu, A.S.; Leswing, K.; Pande, V. MoleculeNet: A benchmark for molecular machine learning. *arXiv* **2018**, arXiv:1703.00564v3. [CrossRef]
10. Walters, W.P.; Barzilay, R. Applications of deep learning in molecule generation and molecular property prediction. *Acc. Chem. Res.* **2020**, *54*, 263–270. [CrossRef]
11. Liao, R.; Zhao, Z.; Urtasun, R.; Zemel, R.S. LanczosNet: Multi-scale deep graph convo-lutional networks. In Proceedings of the 7th International Conference on Learning Representations ICLR 2019, New Orleans, LA, USA, 6–9 May 2019; pp. 1–18.
12. Shindo, H.; Matsumoto, Y. Gated graph recursive neural networks for molecular property prediction. *arXiv* **2019**, arXiv:1909.00259.
13. Shang, C.; Liu, Q.; Chen, K.-S.; Sun, J.; Lu, J.; Yi, J.; Bi, J. Edge attention-based multi-relational graph convolutional networks. *arXiv* **2018**, arXiv:1802.04944.
14. Wang, X.; Li, Z.; Jiang, M.; Wang, S.; Zhang, S.; Wei, Z. Molecule property prediction based on spatial graph embedding. *J. Chem. Inf. Model.* **2019**, *59*, 3817–3828. [CrossRef] [PubMed]
15. Be'Cigneul, G.; Ganea, O.-E.; Chen, B.; Barzilay, R.; Jaakkola, T. Optimal transport graph neural networks. *arXiv* **2020**, arXiv:2006.04804.

16. Mobley, D.L.; Wymer, K.L.; Lim, N.M.; Guthrie, J.P. Blind prediction of solvation free energies from the SAMPL4 challenge. *J. Comput. Aided. Mol. Des.* **2014**, *28*, 135. [CrossRef]
17. Montavon, G.; Rupp, M.; Gobre, V.; Vazquez-Mayagoitia, A.; Hansen, K.; Tkatchenko, A.; Müller, K.-R.; Von Lilienfeld, O.A. Machine learning of molecular electronic properties in chemical compound space. *New J. Phys.* **2013**, *15*, 095003. [CrossRef]
18. Dubey, A. Machine learning model for analysis of critically important antimicrobials for human medicine. *Int. J. Mol. Biol.* **2018**, *3*, 23.
19. Devillers, J.; Domine, D.; Guillon, C.; Karcher, W. Simulating Lipophilicity of Organic Molecules with a Back-Propagation Neural Network. *J. Pharm. Sci.* **1998**, *87*, 1086. [CrossRef]
20. Elton, D.C.; Boukouvalas, Z.; Fuge, M.D.; Chung, P.W. Deep learning for molecular design—A review of the state of the art. *Mol. Syst. Des. Eng.* **2019**, *4*, 828. [CrossRef]
21. Lim, J.; Ryu, S.; Kim, J.W.; Kim, W.Y. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *J. Cheminformatics* **2018**, *10*, 31. [CrossRef]
22. Stahl, M.; Mauser, H.; Tsui, M.; Taylor, N.R. A Robust Clustering Method for Chemical Structures. *J. Med. Chem.* **2005**, *48*, 4358. [CrossRef]
23. Weininger, D. Smiles, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31. [CrossRef]
24. Heller, S.R.; McNaught, A.; Pletnev, I.; Stein, S.; Tchekhovskoi, D. InChI, the IUPAC International Chemical Identifier. *J. Cheminformatics* **2015**, *7*, 23. [CrossRef] [PubMed]
25. Mitchell, B.O. Machine learning methods in chemoinformatics. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2014**, *4*, 468. [CrossRef] [PubMed]
26. Cereto-Massagué, A.; Ojeda, M.J.; Valls, C.; Mulero, M.; GarciaVallvé, S.; Pujadas, G. Molecular fingerprint similarity search in virtual screening. *Methods* **2015**, *71*, 58. [CrossRef] [PubMed]
27. Tang, Y.X.; Lee, Y.H.; Amran, M.; Fediuk, R.; Vatin, N.; Kueh, A.B.H.; Lee, Y.Y. Artificial Neural Network-Forecasted Compression Strength of Alkaline-Activated Slag Concretes. *Sustainability* **2022**, *14*, 5214. [CrossRef]
28. Yakub, I.; Kueh, A.B.H.; Pineda De La O, E.A.; Rahman, M.R.; Barawi, M.H.; Abdullah, M.O.; Amran, M.; Fediuk, R.; Vatin, N.I. Employing an Artificial Neural Network in Correlating a Hydrogen-Selective Catalytic Reduction Performance with Crystallite Sizes of a Biomass-Derived Bimetallic Catalyst. *Catalysts* **2022**, *12*, 779. [CrossRef]
29. ABHYANKAR, A.; Patwardhan, A.; Paliwal, M.; Inamdar, A. Identification of flooded areas due to severe storm using envisat asar data and neural networks. *J. Civ. Eng. Sci. Technol.* **2019**, *10*, 113–120. [CrossRef]
30. ETU, J.E.; Oyedepo, O.J. Forecasting trip generation for high density residential zones of Akure, Nigeria: Comparability of artificial neural network and regression models. *J. Civ. Eng. Sci. Technol.* **2018**, *9*, 2. [CrossRef]
31. Aly, M.; Alotaibi, N.S. A novel deep learning model to detect COVID-19 based on wavelet features extracted from Mel-scale spectrogram of patients' cough and breathing sounds. *Inform. Med. Unlocked* **2022**, *32*, 101049. [CrossRef]
32. Aly, M.; Alotaibi, N.S. A New Model to Detect COVID-19 Coughing and Breathing Sound Symptoms Classification from CQT and Mel Spectrogram Image Representation using Deep Learning. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 601–611. [CrossRef]
33. Yasonik, J. Multiobjective de novo drug design with recurrent neural networks and nondominated sorting. *J. Cheminformatics* **2020**, *12*, 14. [CrossRef] [PubMed]
34. Hirschfeld, L.; Swanson, K.; Yang, K.; Barzilay, R.; Coley, C.W. Uncertainty Quantification Using Neural Networks for Molecular Property Prediction. *J. Chem. Inf. Model.* **2020**, *60*, 3770–3780. [CrossRef] [PubMed]
35. Khoshdeli, M.; Cong, R.; Parvin, B. Detection of Nuclei in H&E Stained Sections Using Convolutional Neural Networks. 2017. Available online: https://www.researchgate.net/publication/314090304 (accessed on 23 May 2022).
36. Xu, Y.; Pei, J.; Lai, L. Deep learning based regression and multiclass models for acute oral toxicity prediction with automatic chemical feature extraction. *J. Chem. Inf. Model.* **2017**, *57*, 2672–2685. [CrossRef] [PubMed]
37. Withnall, M.; Lindelo¨f, E.; Engkvist, O.; Chen, H. Building attention and edge message passing neural networks for bioactivity and physical-chemical property prediction. *J. Cheminformatics* **2020**, *12*, 1. [CrossRef]
38. Yuan, H.; Ji, S. Structpool: Structured graph pooling via conditional random fields. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020; Available online: https://openreview.net/forum?id=BJxg_hVtwH (accessed on 20 March 2022).
39. Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.S.; Leskovec, J. Pre-training graph neural networks. *arXiv* **2019**, arXiv:1905.12265.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.